#4

OIPE
MAY 2 3 2002
JC58
PATENT & TRADEMARK OFFICE

United States Patent

Application Number: 10/078,299

Filing/Receipt Date: 02/20/2002

First Named Applicant: Ying Liu

Confirmation No. 3371

Provisional Application Number 60/296,245

Provisional Filing Date 06/06/2001

Total Claims 28

10078299.062302

Attrasoft Image Retrieval

Inventor: Ying Liu, Savannah, GA

TECHNICAL FIELD

The present invention relates generally to image retrieval and image recognition, and more particularly related to a system, methods, and algorithms of content-based image retrieval and recognition system. Within such a system, the image(s) to be retrieved/recognized is not preprocessed with the association of key words (meta-data). This system allows the user of an image retrieval/recognition system, such as software together with a computer, network server, or web server etc, to define a searching criteria by using an image(s), a segment of an image(s), a directory containing images or combinations of the above. This system will return the result, which contains pairs of the matched image and similarity. The user can see the matched images in a single click.

This invention can be used in image verification (1:1 matching, binary output: yes/no), image identification (1:N matching, single output to indicate a classification), image search or retrieval (1:N matching, multiple output), and image classification (N:1 or N:N matching). For simplicity, we will only use the word, retrieval.

BACKGROUND OF THE INVENTION

In certain types of content-based images retrieval/recognition systems, the central task of the management system is to retrieve images that meet some specified constraints.

Most image-retrieval methods are limited to the keyword-based approach. In this approach, keywords and the images together form a record in a table. The retrieval is based on the keywords in much the same way as the relational database. (Example: Microsoft Access).

The user operation is generally divided into two phases: the learning phase and the search/recognition phase. In the learning phase, various types of processes, such as image preprocessing and image filtering are applied to the images. Then the images are sent to a recognition module to teach the module the characteristics of the image. The learning module can use various algorithms to learn the sample image. In the search/retrieval phase, the recognition module decides the classification of an image in a search directory or a search database.

A very small number of commercially available products exist which perform content-based image retrieval.

Informix Internet Foundation.2000 is an object-relational database management system (ORDBMS), which supports non-alphanumeric data types (objects). IIF2000 supports several DataBlade modules including the Excalibur Image DataBlade module to extend its retrieval capabilities. DataBlade modules are server extensions that are integrated into the core of the database engine. The Excalibur Image DataBlade is based on technology from Excalibur Technologies Corporation, and is co-developed and co-supported by Informix and Excalibur. The core of the DataBlade is the Excalibur Visual retrievalWare SDK. The Image DataBlade module provides image storage, retrieval, and feature management for digital image data. This includes image manipulation, I/O routines, and feature extraction to store and retrieve images by their visual contents. An Informix database can be queried by aspect ratio, brightness, global colour, local colour, shape, and texture attributes. An evaluation copy of IIF2000 and the Excalibur Image DataBlade module can be downloaded from www.informix.com/evaluate/.

IMatch is a content-based image retrieval system developed for the Windows operating system. The software was developed by Mario M. Westphal and is available under a shareware license. IMatch can query an image database by the following matching features: colour similarity, colour and shape (Quick), colour and shape (Fuzzy), colour percentage, and colour distribution. A fully functional 30-day evaluation copy is available for users to assess the software's capabilities and can be downloaded from www.mwlabs.de/download.htm The shareware version has a 2000 limit on the number of images that can be added to a database. A new version of the software was released on the 18th February 2001.

The Oracle8*i* Enterprise Server is an object relational database management system that includes integral support for BLOBs. This provides the basis for adding complex objects, such as digital images, to Oracle databases. The Enterprise release of the Oracle database server includes the Visual Information retrieval (VIR) data cartridge developed by Virage Inc. OVIR is an extension to Oracle8*i* Enterprise Server that provides image storage, content-based retrieval, and format conversion capabilities through an object type. An Oracle database can be queried by global color, local color, shape, and texture attributes. An evaluation copy of the Oracle8*i* Enterprise Server can be downloaded from otn.oracle.com.

SUMMARY OF THE INVENTION

The present invention is different from Informix database where images can be queried by aspect ratio, brightness, global colour, local colour, shape, and texture attributes. The present invention is different from Imatch where images can be queried by colour similarity, colour and shape (Quick), colour and shape (Fuzzy), colour percentage, and colour distribution. The present invention is different from the Oracle8*i* Enterprise Server where images can be queried by color, local color, shape, and texture attributes.

The present invention is unique in its sample image, control process, control parameters, and algorithms. The current algorithms do not use methodologies deployed in the above systems. In particular, the following parameters are not used: aspect ratio, brightness, global colour, local colour, shape, colour similarity, colour and shape (Quick), colour and shape (Fuzzy), colour percentage, and colour distribution, local color, shape, and texture attributes. The present invention has nothing in common with any existing system.

Even the current invention is applied to images, the algorithms in the invention can be applied to other types of data, such as sound, movie, …

1. Process

The present invention is a content-based image retrieval/recognition system, where users specify an image(s) or segment(s); adjust control parameters of the system, and query for all matching images from an image directory or database. The user operation is generally divided into two phases: learning phase and search/recognition phase. In the learning phase, various types of processes, such as image preprocessing, image size reduction, and image filtering are applied to the images. Then the images are send to a recognition module to teach the module the characteristics of the image as specified by an array of pixels. Each pixel is defined by an integer, which can have any number of bits. The learning module can use ABM or APN learning algorithms to learn the sample image. Both the algorithms will be listed in the present invention.

In the search/retrieval phase, the recognition module decides the classification of an image in a search directory or a search database.

In a retrieval/recognition system, a "training" for the system or "learning" by the system is to teach the system what characteristics of an image, or a segment of an image (key) to look for. A system operator completes this step by specifying the sample image(s); specifying the parameters and clicking one button, the "training" button, which appears in the graphical user interface of the system. A "retraining" by the system is to teach the system what characteristics of images to look for, after the system is already trained. Training and retraining together allows the system to learn from many sample image(s) and segment(s) simultaneously.

A "search" or "retrieval" is to look for matching images from an image source such as, directory, many directories, subdirectories, network, Internet, or database, etc. A system operator completes this step by specifying the image source such as search directory(s), specifying the parameters and clicking one button, the "searching" button, which appears in the graphical user interface of the system. The results can be displayed within the software systems or displayed in a program created by the system. Two particular applications are image verification (1:1 matching, binary output: yes/no) and image identification (1:N matching, single output to indicate a classification).

A "classification" or "recognition" is to repeat training and search for each category of images. At the end, a system operator clicks one button, the "classification" button, which appears in the graphical user interface of the system. The results can be displayed within the software systems or displayed in a program created by the system. Classification is an N: N matching with a single output to indicate a classification.

The parameters and settings of a particular operation can be saved and recalled later. Clicking a button, cut and paste, open files, or typing can achieve recalling a saved operation. The saved results are called "batch code". The "Batch" buttons provide means to execute these saved batch codes.

A "process" is a sequence of training and searching, or a classification, or a specification of a batch code and execution of a batch code. They are further divided into a search process, a classification process, and a batch process.

After the operator completes a process, the results consists of a list of pairs; the pairs consist of the matched image and the "weight", which reflects how closely the selected image matches the sample image(s). This list can be sorted or unsorted. This list provides the link to the matched images so the match images can be viewed with a single click.

"System integration" is to combine a software component, which is an implementation of this invention, with an application interface.

The search process, which is applicable to retrieval, verification, and identification, is:

1. Enter key image into the system;
2. Set training parameters and click the training button to teach the system what to look for;
3. Enter search-directory(s);
4. Set search parameter(s), and click the search button;
5. The system output is a list of names and weights:
    - The weight of an image is related to the characteristics you are looking for (the weight is similar to an Internet search engine weight);
    - Click the name of each image and an image will pop up on the screen.

Figure 1 is the flow chart version of this algorithm.

The classification process is:

1. Enter key image into the system;
2. Set training parameters and click the training button to teach the system what to look for;
3. Enter search-directory(s);
4. Set search parameter(s), and click the search button;

5. Repeat the above process for each class and then click the "Record" button. At the end, click the "Classification" button. The output web page will first list the sample images for each class. Then it will list:
   - An image link for each image in the search directory;
   - The classification weights of this image in each search; and
   - The classification of this image as a link.

The batch process is:

1. Provide the batch code to the system, which includes:
   - Click the save button to save the current setting, including key(s), search directory(s), and parameters into a batch code.
   - Click a file button to recall one of the many batch codes saved earlier.
   - Cut and paste or simply type in a batch code by keyboard.
2. Click batch button to execute the code.

An integration process is to combine a software component, which is an implementation of this invention, with an application interface. This invention also specifies a user-graphical-interface for the integration.

2. Parameters

The search, classification, and batch processes require a set of parameters. All the parameters can be specified in the system user interface, either through clicking buttons or through Windows. The parameters are specially related to the ABM and APN algorithms, which will be claimed in this patent.

The "Area of Interest" specifies an image segment, which is specified by 4 numbers: the coordinates of the upper-left corner and the bottom-right corner.

The “internal representation” specifies the dimensions of a pixel array used for computation, which may or may not be the actual image pixel array.

The “Background” or “Background filter” selects an image-processing filter the pixel array must pass through before entering the learning component of the system.

The “Symmetry” represents similarity under certain types of changes, such as intensity, translation symmetry, Scaling, Rotation, oblique, combined rotation and scaling or any combination thereof.

The “Rotation Types” specify the range of rotation if the rotation symmetry is used. Examples are 360°-rotations, -5° to 5° rotations, and -10° to 10° rotations, or other settings that fit the user’s need.

The “Reduction Type” specifies the method used when reducing a large image pixel array to a smaller pixel array.

The “Sensitivity” deals with the sample segment size; high sensitivity is for small segment(s) and low sensitivity is for large segment(s).

The “Blurring” measures the distortion due to data compression, translation, rotation, scaling, intensity change, and image format conversion.

The “Shape Cut” is to eliminate many images that have different shapes from the sample segment.

The "External Weight Cut" is to list only those retrieved images with weights greater than a certain value. The weight Cut is an integer greater than or equal to 0. There is no limit how large this integer can be. The “Internal Weight” Cut plays a similar role as the External Cut in a percent value rather than an absolute weight value.

The “Image Type” specifies the learning component whether to treat the pixel array as black and white images or a color image. It also instructs the learning component whether to use a maximum value, integration, or both.

The "L/S Segment" (Large/Small segment) specifies the system where to focus when searching images.

The “Short/Long” search specifies an image source such as whether to search one directory or many directories.

The “Short Cut” is a Scrollbar to select an integer between 0 and 99; each integer is mapped to a set of predefined settings for the parameters.

The “Border Cut” controls the portions of images to be used in the image recognition.

The “Segment Cut” controls the threshold used to reduce an image into an internal representation.

3. System Layout

Attrasoft Component-Object structure consists of three layers (See Figure 2):

- Application Layer
- Presentation Layer
- ABM Network Layer

The ABM Network Layer has two algorithms to be claimed in the present invention:

- ABM (Attrasoft Boltzmann Machine);

- Attrasoft PolyNet (APN): multi-valued ABM.

This layer is responsible for learning and classification.

The Presentation Layer is an interface between the ABM net layer and the user interface layer. There are two types of data used by the systems: user data or application data, and ABM neural data. ABM networks use ABM neural data. User data depends on the application. The presentation layer converts the image data into neural data used by the ABM layer component.

The Application Layer is the front-end graphical user interface, which the users see directly. This layer collects all parameters required for necessary computation.

4. Algorithms

The ABM layer deploys two algorithms, ABM and APN. The ABM and APN algorithms consist of a combination of Markov Chain Theory and the Neural Network theory. Both theories are well known. The ABM and APN algorithms are newly invented algorithms, which have never been published.

The following terms are well known: Markov chain, state of Markov chain, invariant distribution.

The basic flow chart for ABM and APN algorithms are:

1. Combine an image and its classification into a vector.
2. All such together form a mathematical configuration space. Each point in such a space is called a state.
3. A Markov chain exists in such a space where the state of the configuration space is a state of the Markov chain.

4. The Markov chain will settle on its invariant distribution. A distribution function is deployed to describe such a distribution. In particular, such distribution function classifies the images.
5. The construction of such a Markov chain is by a particular type of neural network, called ABM network or APN network. This type of neural net satisfies 3 features: (1) fully connected; (2) the order of the neural net is the same as the number of neurons in the network, i.e. the number of connections is an exponential function of the number of neurons; and (3) the connections follow particular algorithms, known as ABM and APN algorithms.

The Step 4 of the above is defined as follows:

Let x be an image, and let a, b be two classes; then the two possible vectors are (x, a) and (x, b). Let a distribution function be z = F (y), where y is a vector. If y = (x, a), z = z1; and y = (x, b), z = z2, then the probability of x in class a is z1 and the probability of x in class b is z2. The result will be {(x, a, z1), (x, b, z2)}. The users will see results like this directly in the output of the system.

In the ABM or APN algorithms, content-based image retrieval and image recognition are basically the same problem; therefore, they can be converted from one to the other. To convert from an image search problem to an image recognition problem, one query is required for each class. To see whether an image, say B, is in class A, you first train ABM with all images in class A, then try to retrieve image B. If image B is not retrieved, then image B is not in class A. If image B is retrieved only for class A, then image B is in class A. If image B is retrieved for several classes, the class with the largest relative probability is the one to which image B belongs. Image search is an image classification problem with only 1 class.

ABM is a binary network. APN is a multi-valued network.

5. Components and Application-Programming Interface

Software components can be isolated to be attached to different front-end systems. This can be done with ABM neural layer alone, or both ABM layer and presentation layer. The ABM layer component is a core of the present invention. The value of such a sub-system is the same as the whole system.

This invention also defines the application-programming interface (API), which specifies the system integration. This API is called IVI-API.

BRIEF DESCRIPTION OF VIEWS OF THE DRAWING

Figure 1 shows the algorithm of the Search Process, which is applicable for image verification, identification, and retrieval.

Figure 2 shows a 3-Layer Internal Architecture.

Figure 3 shows a sample User Interface of the Present Invention.

Figure 4 shows a sample Key Input for the Present Invention.

Figure 5 shows a sample Search Output of the Present Invention. The search output is a list of pairs.

Figure 6 shows a sample Classification output of the Present Invention. The classification output is a list of triplets.

DETAILED DESCRIPTION OF THE DISCLOSED EMBODYMENT

Preferred Embodiment of the Search System

An image search/classification constructed in accordance with the preferred embodiment comprises a computer-based workstation including monitor, keyboard and mouse, a content-based image retrieval software system and a source of images.

The source of the images may be on the local drive, network or the Internet. The source is connected to the workstation. The source of images may be accessed directly via open files, or indirectly, such as going into a file to find the images or going into a database application to find the images, etc.

The preferred workstation can be a PC or any other type of computers, which connects to a data source.

The preferred content-based image retrieval software system is any software, which has ABM or APN algorithm as a component. It can be a Window-based system, or any other operating system based systems, or Internet based systems.

Overview of the ABM Algorithm

The following terms are well known: synaptic connection or connection.

The basic flow chart for ABM algorithm is:

1. Create an ABM net with no connections;
2. Combine an image and its classification into an input vector.
3. Impose the input vector to the learning module.

4. The ABM neural connections are calculated based on the input vector. Let N be the number of neurons; the order of connections can be up to N and the number of connections can be 2**N, where ** represent the exponential function.
5. The Markov chain is formed after the connections are established. This Markov chain will settle on its invariant distribution. A distribution function is deployed to describe such a distribution.
6. This distribution function, once obtained, can be used to classify images. This will produce triplets of image, class, and weight. Image retrieval and classification are two different sides of the same token.
7. These triplets of image, class, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The second part of the triple is omitted because the search problem has only one class.

Overview of the APN Algorithm

The basic flow chart for APN algorithm is:

1. Create an APN neural net with no connections;
2. Combine an image and its classification into an input vector.
3. Impose the input vector to the learning module.
4. The APN neural connections are calculated based on the input vector. Let N be the number of neurons; the order of connections can be up to N and the number of connections can be 2**N, where ** represent the exponential function.
5. A mapping over each connection is established. Let K be a number of neurons in a K order connection, where K is less than or equal to N, then this will be a K to K mapping, i.e. the domain of the mapping has K integers and the range of the mapping has K integers.

6. The K-elements mapping is changed to N-element mapping by adding (N – K) pairs of 0 to 0 relations for each of the neurons not in the set K. By taking the domain of this mapping away, the range of this mapping forms a vector, APN connection vector.
7. The Markov chain is formed after the connections are established. This chain will settle on its on its invariant distribution. A distribution function is deployed to describe such a distribution.
8. This distribution function, once obtained, can be used classify images. This will produce triplets of image, class, and weight.
9. Comparing the input-vector and the APN-connection-vector modifies this weight. This will produce a new set of triplets of image, classification, and weight.
10. These triplets of image, class, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The second part of the triple is omitted because the search problem has only one class.

User Interface Layer of software for implementation of ABM and APN Algorithms

There are three major operations:

- Search or retrieval;
- Classification; and
- Batch.

These are the principle modes of the system that runs on the workstation. The software executed in these three modes can have various user interfaces, such as in Windows environment or the web environment, etc. The user interface collects necessary information for the computation.

Other than the key and the a source of images, the user interface may or may not pass the following information to the next layer:

The "Area of Interest" specifies an image segment by two clicks. These two clicks generate 4 numbers the coordinates of the upper-left corner and the bottom-right corner.

The "internal representation" specifies the dimensions of a pixel array used for computation, which may or may not be the actual image pixel array.

The "Background" or "Background filter" selects an image-processing filter the pixel array must pass through before entering the learning component of the system. The interface will be responsible for selecting one of many available filters.

The "Symmetry" represents similarity under certain types of changes, such as intensity, translation symmetry, Scaling, Rotation, oblique, combined rotation and scaling or any combination thereof. For the translation symmetry, this is implemented by physically translating the sample image to all possible positions. The similar methods can be applied to other symmetries.

The "Rotation Types" specify the range of rotation if the rotation symmetry is used. Examples are 360°-rotations, -5° to 5° rotations, and -10° to 10° rotations, or other settings that fit the user's need.

The "Reduction Type" specifies the method used when reducing a large image pixel array to a smaller pixel array.

The "Sensitivity" deals with the sample segment size; high sensitivity is for small segment(s) and low sensitivity is for large segment(s). This is a method to limit the relevant neural connections. When ABM net, x1, is trained, there will be certain connections. All possible connections together form a space, H1. For the ABM net with N neurons, such a space will have a maximum of 2**N point, where ** is the exponential function. Each trained ABM net will have a set h1,

representing non-zero connections. When deciding whether an image, I2, in a search directory is a match to the current sample image, I1, this image I2 can be turned around to train the new but similar ABM neural net, x2. The will generate a set of connections, h2. Similarity determines a maximum distance, d, either using the Hausdorff distance or L1 distance or L2 distance. In the connection space, starting from the connection set, h2, of the new ABM net, after applying this new distance, d, a new set, h3, is obtained. Obviously the smaller this distance, d, is, the smaller this new set, h3, will be. This new set, h3, is then transformed back to h1. Any point in h1 but not in h3 will be considered "too far" and therefore is set to 0 for the current image, I2, in the search directory. This reduction in the connections space is determined by the sensitivity.

The "Blurring" measures the distortion due to data compression, translation, rotation, scaling, intensity change, and image format conversion. This method expands an image in the search directory from a single point to a set as follows. All possible images together form a space, the image space. An image is a point in such a space. When deciding whether an image, I2, in a search directory is a match to the current sample image, I1, this image I2 can be turned a small set around the I2. Let the set be IS2. Blurring determines a maximum distance, d, either using the Hausdorff distance or L1 distance or L2 distance. In the image space, starting from the I2, after applying this new distance, d, a new sphere set, IS2, is obtained. Obviously the smaller this distance, d, is, the smaller this new set, IS2, will be. Now any point in this set, IS2, is just as good as I2. This expansion in the image space is determined by the Blurring.

The "Shape Cut" is to eliminate many images that have different shapes from the sample segment. All possible images together form a space, the image space. An image is a point in such a space. When deciding whether an image, I2, in a search directory is a match to the current sample image, I1, the distance between I1 and I2, d, can be determined, either using the Hausdorff distance or L1 distance or L2 distance. If this distance, d, is larger than a predetermined distance, D, a mismatch can be declared without going through the ABM neural net. This predetermined distance, D, is set by the "Shape Cut" parameter.

The "External Weight Cut" is to list only those retrieved images with weights greater than a certain value. The weight Cut is an integer greater than or equal to 0. There is no limit how large this integer can be.

The “Internal Weight Cut” plays a similar role as the “External Cut” in a percent value rather than an absolute weight value.

The “Image Type” specifies the ABM or APN algorithm. It also instructs the neural layer component how to compute the weights. The weight can be computed by using the invariant function of the Markov chain, or integration all contributions in the time evolution of the Markov chain, with or without reaching the invariant distribution.

The "L/S Segment" (Large/Small segment) specifies the system where to focus when searching images. Please refer to the similarity to understand the set of contributing connections, i.e. not every connection is a contributing connection. Small and Large segments deploy different scales in the determining the set of connections.

The “Short/Long” search specifies an image source such as whether to search one directory or many directories.

The “Short Cut” is a Scrollbar to select an integer between 0 and 99; each integer is mapped to a set of predefined settings for the parameters.

The “Border Cut” is to eliminate the border sections of images. This parameter controls the percentage of images to be eliminated before entering consideration.

The “Segment Cut” is best illustrated by examples. Assume 1 400x400 image is reduced to 100x100 internal representation, as set by the parameter “Internal Representation”; then 16 original pixels will be reduced into 1 pixel. The new value of the single pixel is determined by the parameter “Reduction Type”. The “Segment Cut’ sets a threshold: if the number of non-zero

pixels is greater than the threshold, the pixel will have a non-zero value; otherwise, the pixel will have a zero value.

Presentation Layer of software for implementation of ABM and APN Algorithms

The presentation layer transforms the image data to neural data. The procedure includes:

1. Open files from the image source;
2. Decode the image into pixels arrays;
3. Process images with a filter;
4. Reduce the size of images to an internal representation. The users can arbitrarily choose the internal representation of the images. Such reduction can be based on individual images on a case-by-case reduction, or deploy the same reduction factor across to all images.
5. In the case where many pixels in an image have to be combined into a new pixel before leaving this layer, the user can choose a reduction type such as taking average, maximum, minimum, or deploy a threshold.
6. Pass the image array to the next layer.

ABM Layer of software for implementation of ABM and APN Algorithms

This Upper level of this layer has two branches:

- Training Objects
    - High level training class
    - Low level training class and
    - Symmetry class

- Recognition Objects
    - High level recognition class
    - Low level recognition class

This lower level of this layer has only one class, the memory management class.

The purpose of the memory management class is to claim memory space from RAM, 64K at a time. This memory space will be used for storing the connections. It also returns the unnecessary space back to the operating system of the computer.

The low level training object is to provide all necessary functions used by the high level training class.

The symmetry object is to implement the symmetry defined earlier.

The high level training class incorporates symmetry and implements the ABM or APN algorithm. The "image Type" parameter in the user interface will determine which algorithm will be use.

ABM Training Algorithm is:

1. Delete the existing ABM connections;
2. Combine an image and its classification into an input vector.
3. The ABM neural connections are calculated based on the input vector. Let N is the number of neurons, these connections can be up to the order of N. The image is randomly breaking down into a predefined number of pieces.
4. Let an image piece, p1, have K = (k1 + k2) pixels, where K is an integer. After imposing the pixel vector to the ABM net, k1 is the number of neurons excited and k2 is the neurons of neurons grounded. A neural state vector can be constructed to represent such a configuration, which k1 components being 1 and k2 components being 0.

5. All such vectors together form a space, the connection space. A distance, either the Hausdorff distance or L1 distance or L2 distance can be defined in this space. Such a definition of a distance allows all possible connection vectors to be classified via a distance from p1. Many vectors will be in a group with distance 1 from p1. Many vectors will be in a group with distance 2 from p1, ...
6. The connection represented by p1 is assigned the largest synaptic connection weight. Those connections in the distance 1 group will have smaller weights, ... After a certain distance, the connection weights will be 0, or there will be no connections. The present invention covers all possible combinations of such a generating method.
7. The Markov chain is formed after the connections are established.

APN Training Algorithm is:

1. Delete the existing ABM connections;
2. Combine an image and its classification into an input vector.
3. The ABM neural connections are calculated based on the input vector. Let N is the number of neurons, these connections can be up to the order of N. The image is randomly breaking down into a predefined number of pieces.
4. Let an image piece, p1, have K = (k1 + k2) pixels, where K is an integer. After imposing the pixel vector to the ABM net, k1 is the number of neurons excited and k2 is the neurons of neurons grounded. A neural state vector can be constructed to represent such a configuration, which k1 components being 1 and k2 components being 0.
5. All such vectors together form a space, the connection space. A distance, either the Hausdorff distance or L1 distance or L2 distance can be defined in this space. Such a definition of a distance allows all possible connection vectors to be classified via a distance from p1. Many vectors will be in a group with distance 1 from p1. Many vectors will be in a group with distance 2 from p1, ...
6. The connection represented by p1 is assigned the largest synaptic connection weight. Those connections in the distance 1 group will have smaller weights, ... After a certain distance, the connection weights will be 0, or there will be no connections. The present invention covers all possible combinations of such a generating method.

7. The Markov chain is formed after the connections are established.
8. For each connection, in addition to the synaptic connection weight, a mapping over each connection is established. Let k1 be a number of neurons in the original k1 order connection generated by p1, then this mapping maps from the k1 neuron to the k1 pixel value which excited these neurons. This completes the connection for the original segment p1.
9. The segment, p1, also generated many other connections. If a neuron in this connection is one of the original k1 neurons in p1, then this neuron is mapped into the corresponding pixel value, which causes this neuron to be excited; otherwise, this neurons is mapped into 0. This completes the mappings of all connections generated by this segment p1.

The low-level recognition object is to provide all necessary functions used by the high-level recognition class.

The high-level recognition class implements the ABM or APN algorithm. The "image Type" parameter in the user interface will determine which algorithm will be use.

ABM Recognition Algorithm is:

1. An image to be classified is imposed on the Markov Chain.
2. This Markov chain will settle on its invariant distribution. A distribution function is deployed to describe such a distribution.
3. This distribution function, once obtained, can be used to classify images. This will produce triplets of image, class, and weight. Image retrieval and classification are two different sides of the same token.
4. These triplets of image, classification, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The second part of the triple is omitted because the search problem has only one class.

APN Recognition Algorithm is

1. An image to be classified is imposed on the Markov Chain.
2. This chain will settle on its on its invariant distribution. A distribution function is deployed to describe such a distribution.
3. This distribution function, once obtained, can be used classify images. This will produce triplets of image, class, and weight.
4. Comparing the input-vector and the APN-connection-vector modifies this weight. All connection vectors together forms a vector space. A distance, either L1 distance or L2 distance can be defined in this space. The basic idea is the new weight will be directly proportional to the old weight and inversely proportional to this distance. The present invention covers all functions of obtaining the new weight:

   New weight = f (old weight, distance).

   This will produce a new set of triplets of image, classification, and weight.
5. These triplets of image, classification, and weight can be viewed as the results of the classification process. For the search process, a doublet of image and weight are displayed. The second part of the triple is omitted because the search problem has only one class.

IVI-API (Image Verification and Identification Application Programming Interface)

A typical image matching application structure is:

- GUI (graphical user interface) Layer
- DBMS (database management system) Layer
- IVI-API (image verification and identification API) Layer
- SPI (Service Provider Interface) Layer
- OS (Operating System) and Hardware Layer

The IVI-API is transparent for SPI (Service Provider Interface): the SPI functions will pass right through the IVI-API. The SPI can be accessed directly from layers above the IVI-API layer, i.e. the DBMS layer or GUI layer.

There are two main functions in API layer: verify and identify; and there is one main function in the SPI layer: capture.

The two top-level jobs for verification are Enrollment and Verify. The two top-level jobs for identification are Enrollment and Identify. The enrollment, in either case, is nothing but setting a few parameters; the IVI-API deals with the raw images directly. In this API, there is only one top-level function for verifications, Verify; and there is only one top-level function for identifications, Identify.

This IVI-API does not have an enrollment process. The enrollment is replaced by setting two parameters:

- The image in question;
- The folder of previously stored images.

This IVI-API does require an image storage structure that should be followed by the applications, so the folder of previously stored images can be passed to the verification and identification functions.

Both the verification path and identification path are parameters, which can be changed by the parameter writer functions. The image in question can be stored anywhere in a hard drive. The previously stored images must follow the following structure:

Verification

The previously stored images must be stored at:

verification path\ID\.

Example. Assume:

1. The verification path (a parameter) is:

c:\Attrasoft\verification\

2. A set of doublets is:

Image	imageID
Gina1.jpg	12001
Gina2.jpg	12001
Tiffany1.jpg	12002
Tiffany2.jpg	12002

Then the storage structure is:

c:\Attrasoft\verification\12001\gina1.jpg

c:\Attrasoft\verification\12001\gina2.jpg

c:\Attrasoft\verification\12002\tiffany1.jpg

c:\Attrasoft\verification\12002\tiffany2.jpg

Identification

The folder of previously stored images must be stored at:

identification path\

Example. Assume:

1. The identification path (a parameter) is:

c:\Attrasoft\identification\

2. A set of doublets is:

Image	imageID
Gina1.jpg	12001
Gina2.jpg	12001
Tiffany1.jpg	12002
Tiffany2.jpg	12002

If the number of images is less than 1000, then the storage structure is

c:\Attrasoft\identification\gina1.jpg

c:\Attrasoft\identification\gina2.jpg

c:\Attrasoft\identification\tiffany1.jpg

c:\Attrasoft\identification\tiffany2.jpg

If the number of images is more than 1000, then the sub-directories should be used:

c:\Attrasoft\identification\dir0000\gina1.jpg

c:\Attrasoft\identification\dir0000\gina2.jpg

c:\Attrasoft\identification\dir0000\tiffany1.jpg

c:\Attrasoft\identification\dir0000\tiffany2.jpg

Enrollment

The enrollment process builds the folder of previously stored images according to the above structure. The folder of previously stored images will be a parameter for the AVI layer, called verification directory, or identification directory or search directory. There will be a section to address the parameters later. Because the enrollment means passing parameters, the enrollment is always 100%.

1:N Matching

The following methods (one main function and three result readers) are used to perform the Verification function:

```
int verify(String image, long imageID);
long getVerifyID();
String getVerifyName();
long getVerifyWeight();
```

A typical process is:

- Initialize System

- Capture image
- Calculate the template
- Verify

However, because "Calculate the template" is not required in this IVI-API; and the system is initialized before the verification process started, the process is:

- Capture
- Verify

The capture() functions are provided in SPI, which can be accessed directly by applications. Both the image in question and the folder of previously stored images are in the hard drive. The applications then pass (String image, long imageID) to the verify() function.

N:N Matching

The following methods are used to perform the Identification function:

```
int identify(String image );
Long [] getIdentifyID();
String [] getIdentifyName();
Long getIdentifyWeight().
```

Both the image in question and the folder of previously stored images are in the hard drive. The applications then pass (String image) to the identify() function.

Parameters

The set of parameters forms an array;

```
Void setParameter( int I, long x); // a[I] = x

Long getParameter(int I); // retune a[I]
```

Sample Implementation

We will present three sample implementations based on Figure 2 (3-Layer Architecture). The first example has all 3 layers; the second example has only 1 layer; and the third example has 2 layers.

There are two CD's labeled "Document, Sample Implementation". The disks contain only three **ASCII** files. Each disk in the duplicate set is identical. The contents of the CD are:

File Name	Type	Size	Date	Description
ABM4_9	TXT	156,256	05-16-02	Detailed description of ImageFinder 4.9
ABM5_0	TXT	96,515	05-16-02	Detailed description of PolyApplet 5.0
ABM5_1	TXT	43,019	05-16-02	Detailed description of TransApplet 5.1

These three files will give detailed descriptions of the three sample implementations below.

Attrasoft ImageFinder 4.9

A sample Invention Application Software is the Attrasoft ImageFinder 4.9, which has all three layers in Figure 2. Figure 3 shows the ImageFinder User Interface using the Present Invention. Figure 4 shows a sample Key Input in the ImageFinder software using the Present Invention. Figure 5 shows a sample Search Output of the Present Invention. The search output is a list of pairs. Figure 6 shows a sample Classification output of the Present Invention. The classification output is a list of triplets.

The ASCII file, ABM4_9.TXT, in the CD's labeled "Document, Sample Implementation" will give a detailed description.

In addition, two CD's, labeled "Attrasoft ImageFinder 4.9", contain sample implementation software. The software can be installed and run to test the proposed algorithm. Note:

A. The CD's contain **non-ASCII** files, such as the installation file and execution files. The installation files will install the following executable files to a computer with Microsoft Windows as the operating system:

- Attrasoft ImageFinder 4.9 for Windows 95/98/ME, execution files;
- Attrasoft ImageFinder 4.9 for Windows 2000/XP, execution files;
- Data File for running the software;
- User's Guide in Microsoft Word, and
- User's Guide in html format.

These five files can also be run from the CD.

B. The Operating System is Windows 95, 98, ME, 2000, and XP.

C. Each disk in the duplicate set is identical.

D. Contents of the CD.

Root Directory Contents:

File Name	Type	Size	Date	Description
DISK1	ID	5	01-05-90 9:31p	Installation File
DISK10	ID	5	01-05-90 9:31p	Installation File
DISK11	ID	5	01-05-90 9:31p	Installation File
DISK12	ID	5	01-05-90 9:31p	Installation File
DISK13	ID	5	01-05-90 9:32p	Installation File
DISK14	ID	5	01-05-90 9:32p	Installation File
DISK2	ID	5	01-05-90 9:32p	Installation File
DISK3	ID	5	01-05-90 9:32p	Installation File
DISK4	ID	5	01-05-90 9:33p	Installation File
DISK5	ID	5	01-05-90 9:33p	Installation File
DISK6	ID	5	01-05-90 9:33p	Installation File
DISK7	ID	5	01-05-90 9:33p	Installation File
DISK8	ID	5	01-05-90 9:34p	Installation File
DISK9	ID	5	01-05-90 9:34p	Installation File
SETUP	EXE	47,616	01-05-90 9:31p	Installation File

SETUP	INI	32	01-05-90 9:31p	Installation File
SETUP	INS	147,449	01-05-90 9:31p	Installation File
SETUP	ISS	510	01-05-90 9:31p	Installation File
SETUP	PKG	15,061	01-05-90 9:31p	Installation File
_INST32I	EX_	306,666	01-05-90 9:31p	Installation File
_ISDEL	EXE	8,192	01-05-90 9:31p	Installation File
_SETUP	1	721,623	01-05-90 9:31p	Installation File
_SETUP	10	1,454,681	01-05-90 9:31p	Installation File
_SETUP	11	1,455,574	01-05-90 9:31p	Installation File
_SETUP	12	1,455,468	01-05-90 9:31p	Installation File
_SETUP	13	1,454,113	01-05-90 9:32p	Installation File
_SETUP	14	1,074,165	01-05-90 9:32p	Installation File
_SETUP	2	1,454,796	01-05-90 9:32p	Installation File
_SETUP	3	1,456,887	01-05-90 9:32p	Installation File
_SETUP	4	1,455,245	01-05-90 9:33p	Installation File
_SETUP	5	1,455,918	01-05-90 9:33p	Installation File
_SETUP	6	1,455,206	01-05-90 9:33p	Installation File
_SETUP	7	1,453,720	01-05-90 9:33p	Installation File
_SETUP	8	1,455,603	01-05-90 9:34p	Installation File
_SETUP	9	1,456,571	01-05-90 9:34p	Installation File
_SETUP	DLL	10,752	01-05-90 9:31p	Installation File
_SETUP	LIB	196,219	01-05-90 9:31p	Installation File
ABM49	<DIR>		06-08-01 1:04p	Executable File
USPTO72	<DIR>		02-28-01 7:15p	Data File
USPTO74	<DIR>		05-21-01 4:33p	Data File

E. Interpretation of the files

Please see Appendix A for the detailed interpretation of the roles of these files. To install the software to a Personal Computer using Windows, double click the setup.exe file.

Attrasoft PolyApplet 5.0

A sample Invention Application Software is the PolyApplet 5.0, which only has the Neural Layer of this invention.

The ASCII file, ABM5_0.TXT, in the CD's labeled "Document, Sample Implementation" will give a detailed description.

Attrasoft TransApplet 5.1

A sample Invention Application Software is the TransApplet 5.1, which has both Neural Layer and the Presentation Layer of this invention.

The ASCII file, ABM5_1.TXT, in the CD's labeled "Document, Sample Implementation" will give a detailed description.

In addition, two CD's labeled "Attrasoft TransApplet 5.1" contain sample implementation of the software library. Note:

A. The disks contain only **Non-ASCII** files. The CD contains the following files:

- Attrasoft TransApplet 5.1 software library for Windows 95/98/ME/2000/XP, COM/DLL file format;
- Sample Implementation Code;
- User's Guide in Microsoft Word, and
- User's Guide in html format.

B. The Operating System is Windows 95, 98, ME, 2000, and XP.

C. Each disk in the duplicate set is identical.

D. Contents of the CD:

Root Directory Contents:

File Name	Type	Size	Date	Description
ABM5_1	DOC	616,448	10-21-01 11:28a	User's Guide, Word
CHAP3	<DIR>		10-19-01 4:31p	Examples
CHAP4	<DIR>		10-19-01 4:31p	Examples
CHAP5	<DIR>		10-19-01 4:31p	Examples
CHAP6	<DIR>		10-19-01 4:31p	Examples
CHAP7	<DIR>		10-19-01 4:32p	Examples

FBI	<DIR>		06-08-01 1:04p	Examples
HELP	<DIR>		10-19-01 4:40p	User's Guide, Word
OBLIQUE	<DIR>		06-08-01 1:04p	Examples
README	TXT	567	10-20-01 10:51a	readme.txt
TRANS~26	DLL	282,112	10-21-01 11:00a	COM DLL

E. Interpretation of the files

(E1) The file labeled "COM DLL" is the COM DLL software library file to be used by users.

(E2) The directories, labeled "Examples", contain the examples of how to use the COM DLL.

(E3) The files, labeled "User's Guide, Word" and the directory, "User's Guide, html", contain the User's Guide.